Automatic Facial Expression Analysis and Emotional Classification

by

Robert Fischer

Submitted to the Department of Math and Natural Sciences
in partial fulfillment of the requirements for the degree of a

Diplomingenieur der Optotechnik und Bildverarbeitung (FH)
(Diplom Engineer of Photonics and Image Processing)

at the

UNIVERSITY OF APPLIED SCIENCE DARMSTADT (FHD)

Accomplished and written at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY (MIT)

October 2004

Author . . . .
Department of Math and Natural Sciences
October 30, 2004

Certified by . . . .
Dr. Harald Scharfenberg
Professor at FHD
Thesis Supervisor

Accepted by . . . .
Dr. ing. Bernd Heisele
visiting scientist at MIT

Automatic Facial Expression Analysis and Emotional Classification

by

Robert Fischer

Submitted to the Department of Math and Natural Sciences
on October 30, 2004, in partial fulfillment of the
requirements for the degree of
Diplomingenieur der Optotechnik und Bildverarbeitung (FH)
(Diplom Engineer of Photonics and Image Processing)

Abstract

In this thesis, a system for automatic facial expression analysis was designed and implemented. This system includes a monocular 3d head pose tracker to handle rigid movements, feature extraction based on Gabor wavelets and gradient orientation histograms, and a SVM classifier. Further more, a database with video sequences of acted and natural facial expression was compiled and tests were done, including comparisons to other systems.

Thesis Supervisor: Dr. Harald Scharfenberg
Title: Professor at FHD

Ehrenwörtliche Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der im Literaturverzeichnis angegebenen Quellen und Hilfsmittel angefertigt habe. Die wörtlich oder inhaltlich entnommenen Stellen sind in der Arbeit als solche kenntlich gemacht. Diese Diplomarbeit ist noch nicht veröffentlicht worden und hat daher in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Cambridge, 27. of October

Acknowledgments

Primarily, I thank Bernd Heisele for his encouragement and criticism as well as for all the discussions. My thanks also go to Tomaso Poggio for his invitation to work in his group, and Harald Scharfenberg for supervising this thesis.

Special thanks go to Volker Blanz, Jeffrey Cohn and Alice o'Toole, who provided essential data for this thesis.

I thank Yuri Ivanov for his help and suggestions and Andreas Zumbuehl for proofreading.

I also want to thank Andreas Dreger and Frank Seibert, who stimulated my interest in research, and Ernst Rauscher for his support.

Deutschsprachige Zusammenfassung

Das Erkennen und Interpretieren von Gesichtsausdrücken ist ein wesentlicher Bestandteil unserer täglichen Kommunikation. In der Interaktion mit Machinen fehlt dieses Element jedoch noch völlig. Seit einigen Jahren forschen daher unterschiedliche Gruppen daran, Gesichtsausdrücke automatisch zu erkennen, und so die Kommunikation zwischen Mensch und Maschine menschlicher, aber auch effektiver zu gestalten.

Die meisten der bisher zu diesem Thema veröffentlichten Systeme arbeiten nur unter speziellen Randbedingungen, die einem praktischen Einsatz noch entgegenstehen. Hauptsächlich die Einschränkung auf direkte Frontalansicht in vorgegebener Entfernung von der Kamera widerspricht dem Ziel einer natürlichen Interaktion. Kopfbewegungen, wie sie bei Unterhaltungen üblicher weise vorkommen, führen zu einer Änderung der Perspektive, unter der das Gesicht gesehen wird. Die daraus resultuierende Verzeichnung können viele Systeme nicht ausgleichen.

In dieser Diplomarbeit wurde daher ein Gesichts-Tracker implementiert, der monokular den Ort und die Ausrichtung des Kopfes verfolgt. Bestimmten Punkte im Gesicht, wie zum Beispiel den äusseren Augenecken, sind dabei festen Punkten in einem 3d Modell zugeordnet. In der Bildsequenz werden die Positionen dieser Gesichtsbereiche durch Template-Matching bestimmt und dann das Modell so im Raum gedreht und verschoben, dass bei seiner (Zentral-)Projektion auf die Bildebene die projezierten Punkte auf den im Bild gefundenen Positionen zum liegen kommen. Die Parameter, die die Lage des Modells im Raum beschreiben, werden dabei iterativ mit dem Levenberg-Marquardt Verfahren angepasst.

Mit dieser 3d Information werden die Teile des Gesichtes, die für die Analyse bedeutsam sind, mittels Warping in eine Norm-Perspektive transformiert. Im Gegensatz zu Warping-Verfahren aus der Computergraphik, bei denen eine unverzerrte Textur im Raum projeziert wird, ist in diesem Fall Quell- und Zielbild das Ergebnis einer perspektivischen Projektion. Der funktionelle Zusammenhang dieser Transformation unter Einbeziehung des gegebenen Kopfmodells wurde hergeleitet und eine effektive Berechnungsmethode vorgestellt.

Auf dem so normalisierten Gesicht werden die Koeffizienten für die spätere Klassifizierung ausgerechnet. Zwei Verfahren wurden hierbei verglichen: Gabor Wavelets, die gemäss Neurowissenschaftlern eine bedeutende Rolle in der Bildverarbeitung im Gehirn

spielen, und einfach zu berechnende, lokale Gradienten-Richtungs-Histogramme. Als Klassifikator wurde eine Support Vector Machine gewält.

Zwei frei erhältliche Datenbanken mit Bildern und Videosequenzen von Gesichtsausdrücken wurden auf ihre Eignung als Quelle für Trainings und Testdaten des Systems hin geprüft. Da keine der beiden im Kontext dieser Arbeit als geeignet erschien, entstand die Notwendigkeit, eine eigene Datenbank aufzunehmen. 12 Probanden wurden aufgezeichnet während sie wiederholt bestimmte Gesichtsausdrücke spielten, sowie beim Betrachten kurzer Videoclips, die sie zu natürlichen Gesichtsausdrüken animieren sollten.

In Tests zeigten die Gradienten-Richtungs-Histogramme bessere Ergebnisse als Gabor Wavelets, sowohl in Hinblick auf die Rechenzeit als auch in Hinblick auf die Erkennungsrate. Ein Vergleich mit einem ähnlichen System, basierend auf einer Kombination von Gabor Wavelets und geometrischen Daten, bestätigt dieses Ergebniss.

Contents

List of Figures

List of Tables

Glossary

AU	Action Unit, see appendix B
FACS	Facial Action Coding System, see appendix B
FER	Facial Expression Recognition
GW	Gabor Wavelett, see section 3.1
GOH	Gradient Orientation Histogram, see section 3.2
HMM	Hidden Markov Models, classifier based on probability density distribution
LFA	Local Feature Analysis, feature extraction method [PA96]
PCA	Principal Component Analysis, feature space reduction method based on eigenvalues
NCC	Normalized Cross Correlation, often used for template matching
ROI	Region of Interest, the part of the image on which the processing is limited
SVM	Support Vector Machines, a type of classifier, explained e.g. in [HTF01]

Chapter 1

Introduction

The detection of faces and the interpretation of facial expression under varying conditions is an everyday task for humans, which we fulfill without effort. The identity, age, gender as well as the emotional state can be seen from someones face. The impression we get from a displayed expression will affect our interpretation of the spoken word and even our attitude towards the speaker himself. Humor and sympathy are just two examples for essential informations that are primarily communicated via facial expressions. Hence, they have a high importance for our daily life even though we often are not aware of it.

For computer based systems on the other side, it still is hard to open up this very important channel of communication. Progress in this field promises to equip our technical environment with means for more effective interaction with humans and hopefully one day with something like tact.

1.1 What are Facial Expressions?

Fasel and Luttin define facial expressions as temporally deformed facial features such as eye lids, eye brows, nose, lips and skin texture generated by contractions of facial muscles. They observed typical changes of muscular activities to be brief, "*lasting for a few seconds, but rarely more than five seconds or less than 250 ms.*" [FL03] They also point out the important fact that felt emotions are only one source of facial expressions besides others like verbal and non-verbal communication or physiological activities.

Though facial expressions obviously are not to equate with emotions, in the computer vision community, the term "facial expression recognition" often refers to the classification of facial features in one of the six so called basic emotions: happiness, sadness, fear, disgust, surprise and anger, as introduced by Ekman in 1971 [EF71]. This attempt of an interpretation is based on the assumption that the appearance of emotions are universal across individuals as well as human ethnics and cultures.

1.2 Automatic Facial Expression Analysis

The task of automatic facial expression analysis can be divided into three main steps: face detection, facial feature extraction and classification into expressions.

Detecting a face in a complex scene is nontrivial problem. Most of the existing systems for expression analysis require the face to be in frontal view under defined conditions. For these systems, usually information about the presence of a face and its coarse location in the scene has to be given. Still, the exact location, scale and orientation have to be determined by hand or by an automatic tracking system. Head motion, occlusion, changing illumination and variations in facial features (e.g. closing of eyes) complicate the situation. The human brain is highly trained for this task, therefore we find and analyze faces effortlessly under almost any conditions with a minimum of information, sometimes even when there is no face at all, like in cloud or rock pattern. The presence of hair, glasses or jewelery seems to be no problem for the human visual system, whereas automatic approaches are easily mislead.

After localizing the face, as much information as possible about the displayed facial expression has to be extracted. Several types of perceptual cues to the emotional state are displayed in the face: relative displacements of featured (e.g. raised eyebrows), quasi textural changes in the skin surface (furrowing the brow), changes in skin hue (blushing) and the time course of these signals [LAKG98]. Depending on how the face and its expression are modeled, features have to be designed that condense this information or a part of it to a set of numbers building the base for the classification, and therefore primarily deciding about the quality of the final analysis result.

Most automatic facial expression analysis systems found in the literature directly classify in terms of basic emotions. This is an attempt of interpretation rather than the classification of really observed facial appearance. Some research groups therefore follow the idea of Ekman and Friesen [EF78] who, in the late 70ies, postulated a system that categorizes all possible, visually detectable facial changes in 44 so-called Action Units (AUs). This system, known as Facial Action Coding System (FACS) has been developed to facilitate objective measurements of facial activity for behavioral studies. The interpretation of the AUs in terms of basic emotions then is based on a special FACS dictionary (for additional information please see appendix B).

FACS are an important tool in behavioral science, and the underlying study can be seen as the theoretical basis for any facial expression analysis. Nevertheless, the AU coding is skipped in most Human Computer Interaction (HCI) applications, because it contributes little to the goal of interpreting nonverbal feedback from a user.

Classification is complicated by the fact that despite cross cultural similarities, facial expressions and the intensity with which they are exhibited strongly vary between individuals. Also, it is doubtful that naturally expression can be unambiguously classified into one of the six basic categories. Quite often, facial expressions are blended and their interpretation mainly depends on the situational context. Automatic classification furthermore is confronted with a physiognomic variability due to gender, age and ethnicity.

1.3 Related Work

Facial Expression analysis dates back to the 19th century when Bell [Bel96] and Darwin [Dar72] studied the anatomical and physiological basis of facial expressions of man and animal. Since the mid 1970s, automatic facial expression analysis has attracted the interest of many computer vision research groups. Early works (before the mid 1990s) often perform automatic facial data extraction by using facial motion analysis.

Samal and Iyengar [SP92], Pantic and Rothkrantz [PR00] and more recently Fasel and Luettin [FL03] present good surveys about the different approaches that where made to meet the challenges that come along with facial analysis. Donato et al. [DBH$^+$99] concentrates on systems for automatic FACS-coding and presents an interesting comparison study of different feature extraction technologies.

Table 1.1 lists some of the common methods that are used in various combinations. Designations may differ from other publications, as there is no unified nomenclature defined in this field.

face detection & tracking	feature extraction	classifier
template matching	Gabor wavelets	SVM
facial feat. shape (e.g. iris)	feature point displacement	HMM
eigen faces (PCA)	optical flow	backprop. network
color histogram	labeled graph	expert rules
manually	eigen faces (PCA)	
IR retina reflex	difference images	
	LFA	

Table 1.1: Methods and technologies used for FER

1.4 Aim of this Research

Facial expression analysis is applied either to behavioral science or human computer interaction. This work focuses on the latter subject. Therefore we assume a situation where the interest of one single person at a time is addressed toward the camera or a close-by object e.g. a monitor. This means the person would not turn his head in a position from where he himself cannot observe the object of interest any more. Nevertheless, head motions have to be expected and handled by the system. Although facial expressions differ between individuals, the system is expected to be able to generalize. An ideal approach would automatically detect a face and recognize facial expressions independent of gender, age or ethnicity in real time with a minimal effort of technical hardware.

In this thesis, a system was aimed that would track the head pose to normalize the face image for feature extraction by warping it to canonical frontal view, but also to use the pose as an additional source of information for the classification (the latter could not be done within the given period of time). Feature extraction was planned based on Gabor wavelets with a SVM as classifier.

Chapter 2

Pose Tracker

A common drawback of existing FER systems is their high demands for input data: most of them require frontal view and do not allow for even small head motions. Natural emotions, on the other hand, usually go along with head movements that include translations and rotation in all three dimensions.

Using a stereo camera system would considerable simplify head pose tracking, stabilize it, and reduce the effects of partial occlusion. However, in a real world application a monocular approach is highly desirable since a stereo system not only increases the technical but also the financial effort.

2.1 Related Work

A large number of studies exist on tracking using optical flow. Recently, deformable model based approaches were explored by Eisert et al. [EG98], Gotkur et al. [SGG02] and others. Gotkur et al. also applied their tracker to facial expression recognition. They use a model consisting of 19 points to obtain the differential of pose and deformation between two frames. Although the tracking algorithm itself works in a monocular mode, prior to tracking every new face has to be trained with stereo data to compute an average shape for a person. Needless to say that this fact not only limits the algorithm in many ways concerning real world scenarios but also requires additional hardware.

Loy et al.[LHO00] presented a 3d head tracker for their automated lipreading system. They needed to track the speakers head movements (esp. rotations) in order to compensate for the effect the viewing angle has on the mouth shape. Their solution was based on an algorithm developed by Lowe [Low91] for model-based motion tracking of rigid objects. While tracking face features by a standard template matching method, they solved for the projection and model parameters that best fit the 3d model with respect to the found feature positions. Selected features were the outer corner of each eye and the inner corner of one nostril. The authors reported robust tracking for out-of-plane head rotations of up to 30 degrees.

Braathen et al [BBLM01] estimate for each subject the 3d head geometry and camera parameters from a set of 30 images with 8 manually labeled facial landmarks. For this 3d head model the most likely rotation and translation parameter in each video frame is then found with stochastic particle filtering (also known as Markov Chain Monte-Carlo method). Finally the model that fits best is textured with the image data of the current frame, rotated to frontal, and then projected back to the image plane. Besides the intensive manually preparation for each subject, this approach is computationally expensive since 100 particles were used for filtering and the whole face is warped to frontal view.

2.2 A Monocular 3d Model-Based Pose Tracker

The pose tracker presented in this thesis is the adaptation of the idea introduced by Loy et al. [LHO00] to facial expression recognition. Since facial features undergo strong deformations during facial expressions, it was necessary to extend this approach in different ways like e.g. by a two-step search and feature warping. Warping is limited to small facial regions that carry high information about the expression.

2.2.1 Pose Estimating According to Lowe

The Pin Hole Camera Model

The approach of Lowe is based on the the pin hole camera model. The idea is to approximate the model parameters from a given estimated projection situation that will best fit to the feature positions found in the image. Using the pin hole camera model, the projection of a 3d model point $\vec{a}$ onto the image plane (not considering distortion) can be described as follows:

$$\vec{b} = T \cdot \vec{a} \tag{2.1}$$

$$u_x = f\frac{b_x}{b_z} \qquad u_y = f\frac{b_y}{b_z} \tag{2.2}$$

where T is the transformation matrix in homogeneous coordinates. T performs the rotation around the principle axis with the angles α, β and ϕ and a translation in x, y and z.

$$T = M(x, y, z) \cdot R_z(\phi) \cdot R_y(\beta) \cdot R_x(\alpha) \tag{2.3}$$

The factor f represents the effective focal length. The image coordinates in pixels can be calculated by

$$q_x = \frac{u_x}{s_x} + c_x \qquad q_y = \frac{u_y}{s_y} + c_y \tag{2.4}$$

with $\vec{s}$ being the pixel pitch, that is the physical distance of two pixel centers on the sensor chip, and $\vec{c}$ the coordinates of the pixel where the optical axis hits the sensor.
To reduce the number of parameters, $\vec{c}$ can be approximated by the image center and the effective focal length, as well as the pixel pitch, can be estimated in advance, if not known from the experimental setup respective the specifications of the camera. The parameter set left to be computed therefore is (in the following, we will refer to this parameter set $\vec{p}$ as ”pose”):

$$\vec{p} = \{x, y, z, \alpha, \beta, \phi\} \tag{2.5}$$

Parameter approximation

Lowe [Low91] points out, though the projection from 3D to 2D is a nonlinear operation, it is ”*a smooth and well-behaved transformation*”. Considering small changes between a known pose and another, unknown one (like e.g. in the case of pose tracking), or given a appropriate initial start pose, the function can be considered to be locally linear. With this assumption, the pose parameters can be approximated iteratively.

$$\vec{p}_{i+1} = \vec{p}_i - \vec{d} \tag{2.6}$$

In every iteration step, a correction vector $\vec{d}$ is computed to minimize a vector $\vec{e}$ of error measurements, consisting of the distances between the projected model points and the found positions of the corresponding features in the image. Applying Newton’s method (with J

being the Jacobean matrix), $\vec{d}$ can be found as follows:

$$J\vec{d} = \vec{e} \tag{2.7}$$

This can be solved by computing the pseudo inverse.

$$\vec{d} = (J^tJ)^{-1}J^t\vec{e} \tag{2.8}$$

This solution brings up a problem, because every parameter is treated equally, though a rotation of e.g. 0.5 rad (28.6°) might result in a larger change in the projection situation than e.g. a shift of 50 mm. To normalize the solution, it is therefore necessary to weight each row of the matrix equation according to it's standard deviation. This is done by a diagonal matrix W in which each element is inversely proportional to the parameters standard deviation σ

$$W_{ii} = \frac{1}{\sigma_{p_i}} \tag{2.9}$$

To force convergence, a scalar parameter λ is added controlling the weight of stabilization. The resulting equation is also known as the Levenberg-Marquardt method [Lev44, Mar63]:

$$\vec{d} = (J^tJ + \lambda W^tW)^{-1}J^t\vec{e} \tag{2.10}$$

Since W is a diagonal matrix, adding λW^tW simply means adding constants to the diagonal of J^tJ. Obviously, the largest computational costs lie in computing the Jacobean matrix J.

This algorithm was simulated and tested in Maple for various projection situations before being implemented in C++.

2.2.2 3d Head Model

The algorithm given above was designed for 3d tracking of rigid objects. The face on the other hand, especially when expressing emotions, is (fortunately) not rigid at all. This simple fact makes the goal of developing a robust tracker a real challenge, since the 2d feature tracking (in the image sequence) is done by template matching and the used 3d model defined rigid.

The head model itself, that is the list of 3d coordinates for every facial landmark, was supplied by Volker Blanz. From his database, containing over 200 3d laser scanned faces, he generated an average head and sent us the coordinates of the points marked in figure 2-1 including the normal vectors on the surface at these points. This normal vector was later used for warping operations as explained in section 2.2.3.

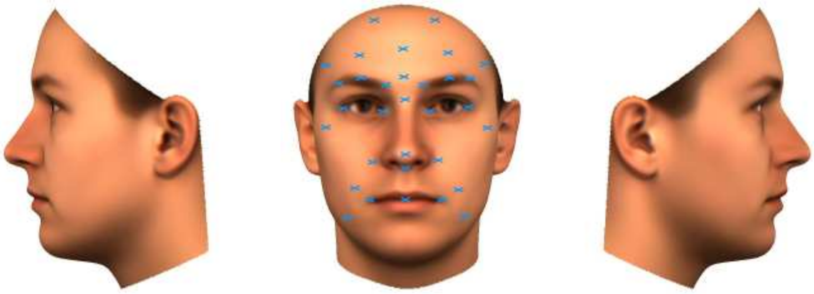

Figure 2-1: 3d Head Model, reduced to the points marked in the center

One might think that since the algorithm above is a regression method, the problems coming along with a non-rigid object would easily be solved by taking more features into account. A lot of experiments were done following this idea, trying to use eye brows, whole eyes, both nostrils separately, inner eye corners, the upper lip and whatever point else can be found in a regular face. There were even attempts to acquire dynamically new features to track by using points with high local variance, and estimate resp. interpolate their corresponding point in the 3d head model. All of this was rejected for one or more of the following reasons:

- the relative position of the feature point on the face and / or its appearance was affected by the facial expressions or movements (e.g. mouth, pupils).
- the normalized cross correlation answer was ambiguous (e.g. eye brows)
- the feature often was affected by occlusion (e.g. inner eye corner)
- the feature frequently disappears (pupils)

The most stable features turned out to be the nostrils, when taken together, followed by the outer eye corners. This was tested by tracking the single features in sequences of different subjects. This set of three features performed significantly better than any other tested set and was therefore used to implement the tracker. Adding other features only resulted in a decrease in performance.

To speed up the search and to stabilize the tracker, a two-step search was used: First, with the last found pose, the positions of the larger features such as each eye with its brow and the nose were calculated by projecting their 3d model point to the image plane. In a downsampled image (by a factor of 1/4) they then were searched and a rough pose estimation was done by the algorithm presented above (see 2.2.1). With this pose, the positions of the detailed features were calculated analogue by projecting their 3d model point to the image plane. The detailed features now were searched in a small ROI around their estimated position in full resolution. The position of the maximum correlation of each detailed feature then was used to apply the parameter approximation again, just like before for the rough pose estimation. To smooth the tracked movement, a momentum parameter κ was applied to the found pose parameters:

$$p_{t+1} = \kappa \; p_{found} + (1 - \kappa) \; p_t \tag{2.11}$$

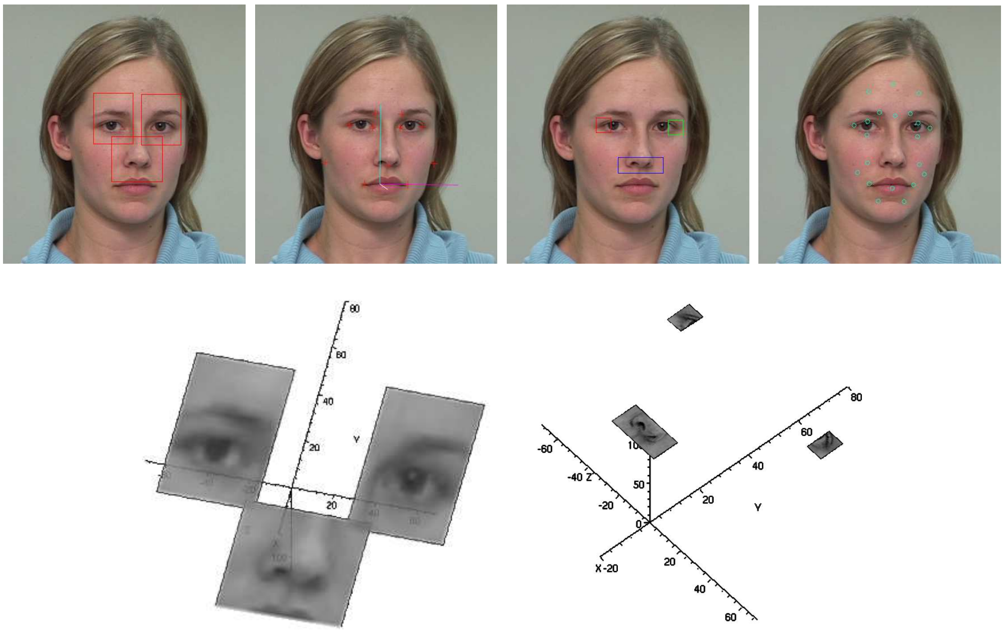

Figure 2-2: Model building

Figure 2-2 shows the coarse and the detailed model. To build them, initially the positions of the detailed features have to be marked manually in the first frame. The templates are then automatically cut off, the pose is calculated and saved in form of the transformation matrix T_{p_0} (used later for warping, see equation 2.13). Thereafter for the rough features the projection of the feature points to the image plane is calculated, and these features are also cut out automatically, this time from the down sampled image. Templates are saved as images files to be loaded for other sequences of the same person.

Though this figure like others later are in color, any processing including the tracking was exclusively done on gray value images.

2.2.3 Feature Update by Warping

Along with their positions, tracked features change their appearance with the change of head poses. For instance a head movement towards or away from the camera would result in a scaling, an in-plane rotation in a rotation and an out-of-plane rotation in a distortion of the template. This effect can be compensated with the information won by the tracking. All that has to be done is warping the original template according to the change in pose. Furthermore, the same effects of distortion affect the features used to extract the data for the classification. A good warping algorithm therefore is required for both tracking and feature extraction.

In our application warping is done on small patches around points of the 3d model, either for tracking or for feature extraction. This approach significantly reduces the computational costs compared to a warping of the whole face.

Image warping is a geometric transformation which smoothly deforms an image, in this case to calculate how a part of the face would look like under another head pose (camera fixed). In more mathematical terms, the transformation of a source image into a destination image is done according to a mapping between the source space (v_x, v_y) and a destination space (k_x, k_y). The mapping usually is specified 'backwards' by the function $\vec{v} = g(\vec{k})$, which practically means to go through the destination image and search for every pixel $\vec{k}$ the corresponding 'origin' pixel $\vec{v}$ in the source image. Since $\vec{v}$ will be a point with real coordinates in the discrete pixel plane, interpolation is required. For this thesis bilinear interpolation is used.

The algorithm developed here basically is a projective mapping (also known as perspective or homogeneous transformation) although it differs from the standard algorithm used in computer graphics in a small but crucial point: For computer graphics an undistorted texture is first mapped to a 3d space by rotation and translation, and afterward projected (P is the projection matrix in homogeneous coordinates) to the destination plane.

$$\vec{k} = P \cdot T \cdot \vec{v} \tag{2.12}$$

In the case of the head tracker, the source image already is a projection under a different pose $\vec{p_0}$. We therefore have to consider an additional translation, rotation and projection:

$$\vec{k} = P \cdot T_{p_1} \cdot T_{p_0}^{-1} \cdot P^{-1} \cdot \vec{v} \tag{2.13}$$

Since a projection is not a bijective mapping, it is obvious, that this equation can only be solved up to scale. Additional constraints are needed for an exact solution. Approximating the surface around the the processed feature point as plane supplies the necessary condition.

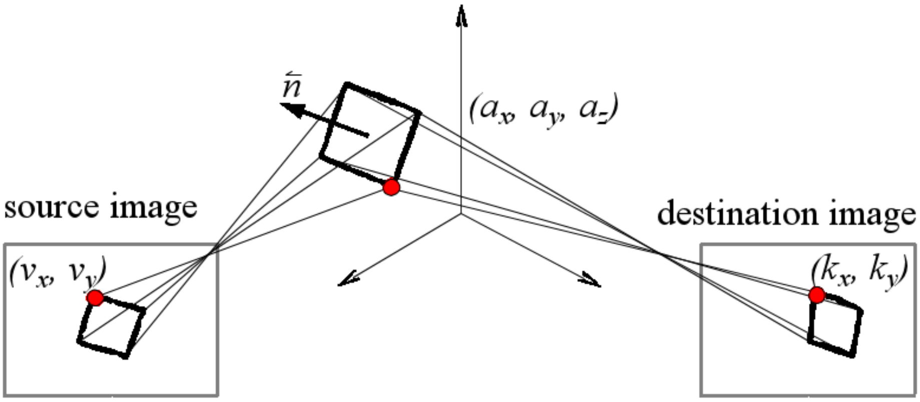

Figure 2-3: Sketch of warping situation

In a nutshell, to get the function $\vec{v} = g(\vec{k})$ we first solve equation 2.13 for $\vec{k} = g^{-1}(\vec{v})$ and then invert it. Figure 2-3 shows, how every $\vec{k}$ is connected to a $\vec{v}$ over a point in the 3d space, we will call $\vec{a}$. To recover the information lost with the projection, we introduce the constraint that $\vec{a}$ has to lie on plane defined through the 3d model point for this patch and the normal vector on it.

With this plane we approximate the skin surface at this region (the patch size ranges between one and two iris) on the face. It can be described in Hesse form as

$$d = \langle \vec{a}, \vec{n} \rangle \tag{2.14}$$

Solved for a_z this is

$$a_z = \frac{1}{n_z}\left(d - a_x n_x - a_y n_y\right) \tag{2.15}$$

and $\vec{a}$ can be written as (now in homogeneous coordinates)

$$\vec{a} = \begin{pmatrix} a_x \\ a_y \\ \frac{d}{n_z} - \frac{a_x n_x}{n_z} - \frac{a_y n_y}{n_z} \\ 1 \end{pmatrix} \tag{2.16}$$

Equipped with $\vec{a}$, equation 2.13 can be broken in two parts:

$$\vec{v} = P \cdot \vec{a} \tag{2.17a}$$

$$\vec{k} = P \cdot T_{p_1} \cdot T_{p_0}^{-1} \cdot \vec{a} \tag{2.17b}$$

where $\vec{v}$ is $\vec{a}$ projected to the source image plane

$$v_x = \frac{f n_z a_x}{d - a_x n_x - a_y n_y} \tag{2.18a}$$

$$v_y = \frac{f n_z a_y}{d - a_x n_x - a_y n_y} \tag{2.18b}$$

Changing now the direction we go through the term 2.13, $\vec{a}$ can be written as expression in $\vec{v}$ (after solving equation 2.18 for v_x and v_y):

$$\vec{a} = \begin{pmatrix} \frac{v_x d}{v_x n_x + v_y n_y + f n_z} \\ \frac{v_y d}{v_x n_x + v_y n_y + f n_z} \\ \frac{f d}{v_x n_x + v_y n_y + f n_z} \\ 1 \end{pmatrix} \tag{2.19}$$

The transformations for both poses can be combined to S which not only shortens the mapping equation but also reduces the computational costs since this operation needs to be done only once in advance.

$$S = T_{p_1} \cdot T_{p_0}^{-1} = \begin{pmatrix} s_{11} & s_{12} & s_{13} & s_{14} \\ s_{21} & s_{22} & s_{23} & s_{24} \\ s_{31} & s_{32} & s_{33} & s_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.20}$$

$$\vec{k} = P \cdot S \cdot \vec{a} \tag{2.21}$$

Analogue to $\vec{v}$, next the projection of $\vec{k}$ to the destination image plane is calculated:

$$k_x = \frac{f\left(s_{11}v_xd + s_{12}v_yd + s_{13}df + s_{14}\left(v_xn_x + v_yn_y + fn_z\right)\right)}{s_{31}v_xd + s_{32}v_yd + s_{33}df + s_{34}\left(v_xn_x + v_yn_y + fn_z\right)} \tag{2.22a}$$

$$k_y = \frac{f\left(s_{21}v_xd + s_{22}v_yd + s_{23}df + s_{24}\left(v_xn_x + v_yn_y + fn_z\right)\right)}{s_{31}v_xd + s_{32}v_yd + s_{33}df + s_{34}\left(v_xn_x + v_yn_y + fn_z\right)} \tag{2.22b}$$

We now have the function $\vec{k} = g^{-1}(\vec{v})$. But what we really want is the inversed expression $\vec{v} = g(\vec{k})$. Therefore equation 2.22 is solved for v_x and v_y. The result will look a bit untidy in the first moment, but cleaning it up, we can identify parts that contain either k_x, k_y or none of them. These factors again need to be computed only once in advance to reduce the overall computation costs.

$$\begin{aligned} \Phi &= k_x\phi_1 + k_y\phi_2 + \phi_3 \\ \phi_1 &= n_x(s_{24}s_{32} - s_{22}s_{34}) + n_y(s_{21}s_{34} - s_{24}s_{31}) + d(s_{21}s_{32} - s_{22}s_{31}) \\ \phi_2 &= n_y(s_{12}s_{34} - s_{14}s_{32}) + n_z(s_{14}s_{31} - s_{11}s_{34}) + d(s_{12}s_{31} - s_{11}s_{32}) \\ \phi_3 &= f(n_y(s_{14}s_{22} - s_{12}s_{24}) + n_z(s_{11}s_{24} - s_{14}s_{21}) + d(s_{11}s_{22} - s_{12}s_{21})) \end{aligned} \tag{2.23}$$

$$\begin{aligned} v_x &= \frac{1}{\Phi}\left(k_x\gamma_1 + k_y\gamma_2 + \gamma_3\right) \\ \gamma_1 &= n_y(s_{24}s_{33} - s_{23}s_{34}) + n_z(s_{22}s_{34} - s_{24}s_{32}) + d(s_{22}s_{33} - s_{23}s_{32}) \\ \gamma_2 &= n_y(s_{13}s_{34} - s_{14}s_{33}) + n_z(s_{14}s_{32} - s_{12}s_{34}) + d(s_{13}s_{32} - s_{12}s_{33}) \\ \gamma_3 &= f(n_y(s_{14}s_{23} - s_{13}s_{24}) + n_z(s_{12}s_{24} - s_{14}s_{22}) + d(s_{12}s_{23} - s_{13}s_{22})) \end{aligned} \tag{2.24}$$

$$
\begin{aligned}
v_y &= \frac{1}{\Phi}\left(k_x\eta_1 + k_y\eta_2 + \eta_3\right) \\
\eta_1 &= n_x(s_{24}s_{33} - s_{23}s_{34}) + n_z(s_{21}s_{34} - s_{24}s_{31}) + d(s_{21}s_{33} - s_{23}s_{31}) \\
\eta_2 &= n_x(s_{13}s_{34} - s_{14}s_{33}) + n_z(s_{14}s_{31} - s_{11}s_{34}) + d(s_{13}s_{31} - s_{11}s_{33}) \\
\eta_3 &= f(n_x(s_{14}s_{23} - s_{13}s_{24}) + n_z(s_{11}s_{24} - s_{14}s_{21}) + d(s_{11}s_{23} - s_{13}s_{21}))
\end{aligned} \tag{2.25}
$$

Both $\vec{k}$ and $\vec{v}$ used for the calculations above are in real world coordinates on their image planes, practically that is the sensor chip. To transform them in pixel coordinates of the digitized images simply equation 2.4 has to be applied.

The development and testing of this algorithm in Maple took about a week, the implementation in C++ not even one day, and finding the one '-' that did not belong where it was set, again about a week.

2.2.4 Qualitative Tracking Results

According to the complexity of the tracker, the tracking results disappoint. Though it performed sufficient for the frontal view sequences in the database, the author expected more stability and accuracy in reward for his effort and patience.

The main problem results from the fact that the face, especially when expressing emotions, is anything but a rigid object, whereas the algorithm originally was designed for rigid objects with a multitude of stable features to track (e.g. corners). Only three features where found to supply the necessary stability, as explained in section 2.2.2.

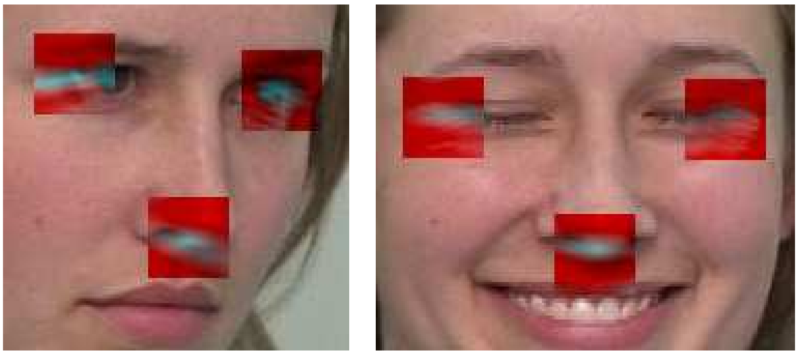

Figure 2-4: The NCC results blended in original image

Figure 2-5 shows some later frames of the same sequence from which the model above was taken (first frame). Even strong noise due to the interlaced effect like in frame 81 or 216 seems to be less problematic than closed eyelids. Frame 87, 174 and 223 show the typical behavior when one detailed feature like here the outer right eye corner is not found. In this case the tracker uses the position of the corresponding coarse feature witch is not found precisely enough due to the changed appearance of the closed eye compared to the template with the open eye. How weak the signal of the NCC is under these conditions can be seen in figure 2-4, where cyan signifies high coefficients and red low coefficients. Reducing the minimum correlation limit is not a solution, since with eye blinking the outer eye corner often could be observed tending toward the nose along the line of the closed eye lid.

The sequence consists of a spontaneous natural expression with fast head movements. Hence the tracked pose is smoothed as declared in equation 2.11, the tracker did not always follow fast enough, as e.g. between the frames 81 and 82. With a higher value for κ on the other hand, the pose would slightly tremble.

The program marks the feature positions with small circles as shown in frame 26. For a better visualization the points in the other frames are remarked at the original positions with green dots.

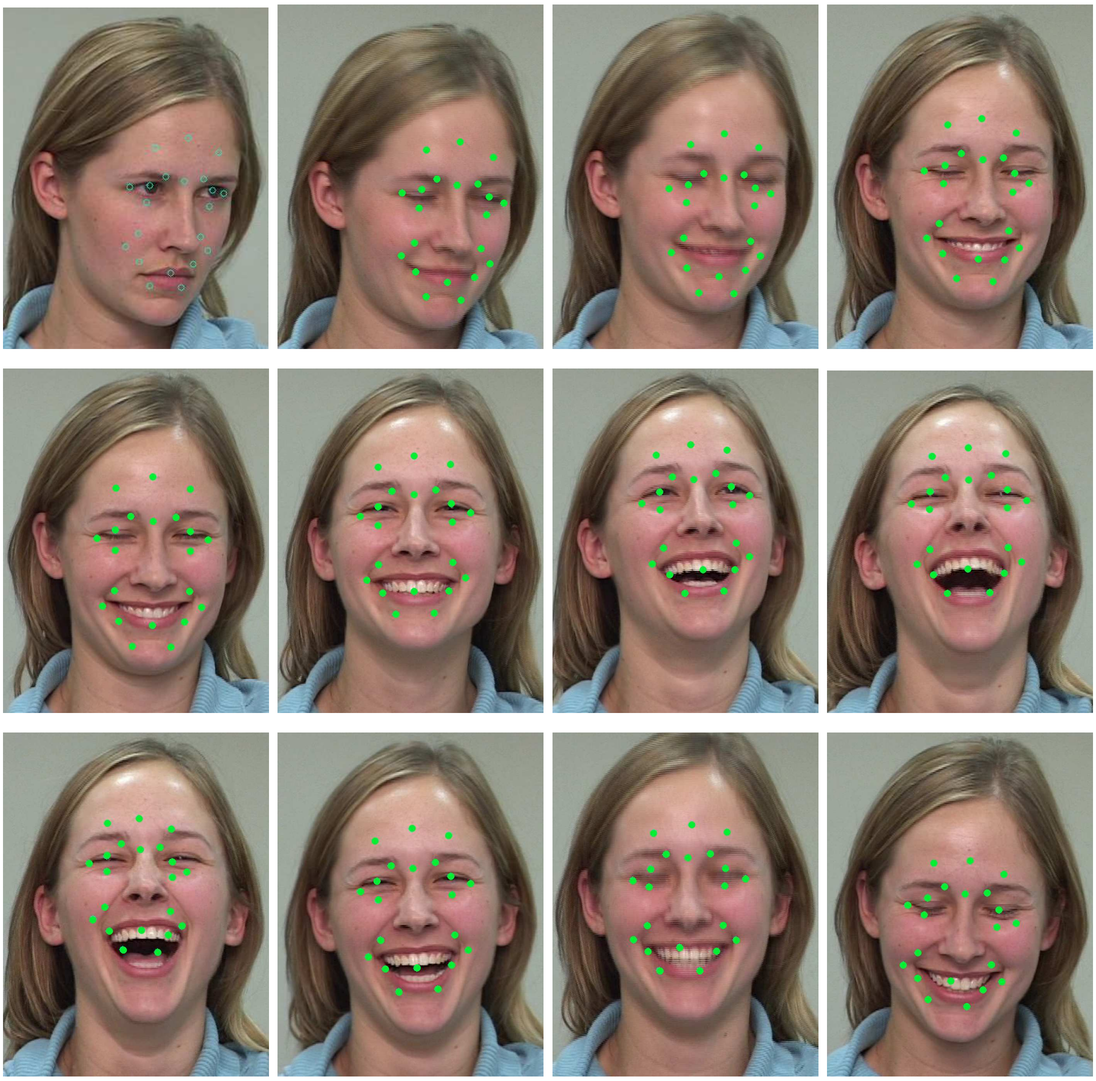

Figure 2-5: Tracking results. Frame numbers: (26, 81, 82, 87), (92, 119, 161, 168), (174, 194, 216, 223)

Chapter 3

Feature Extraction and Classification

Facial feature extraction attempts to find the most appropriate representation of the source data, that is the image or sequence with the expression to recognize. The goal is to condense the data to a minimum, not only to reduce the amount of data to be processed, but also to extract the information meaningful regarding the task at hand. Many different kind of features and their combinations have been tested by various groups like e.g. geometrical features [KB03, KQP03, YTC02, PR98], blob features [OPB97], optical flow [EP95, OO98], active shape models [FT03, LTC97] and Gabor wavelets (GW).

The choice of the feature primarily determines the recognition quality and computational effort. Hence several comparative studies (e.g. [DBH+99, YTC02, FL03, BML+]) indicate that GWs are outperforming other features, they have been selected and implemented in this thesis. In addition, a much simpler type of features was tested and compared to GW.

After feature extraction, classification is the last step of an automatic FER. Here too, many approaches were explored by different research groups. Although a general comparison is hard to do, the same studies as above show that systems running SVM [BLFM03] classifiers usually perform better than others.

3.1 Gabor Wavelets (GW)

1998 Lyons et al [LAKG98] proposed a feature extraction based on Gabor Wavelets motivated by a "significant level of psychological plausibility". Simultaneously Zhang et al.

compared them with geometry-based features using a multi-layer perceptron with good results for the GWs. Since then, many automatic facial expression recognition systems were based or additionally equipped with GWs [BLB$^+$03, BKP03, SGG02, YTC02].

Lyons and Zhang adapted for their systems the idea of the von der Malsburg group that designed a face recognition system using GWs jets together with an elastic bunch graph matching [LVB$^+$, WFKvdM99]. This system gained some fame because it clearly outperformed other face recognition system at the time. However, the root of this idea of applying GW on recognition tasks can be tracked even further back to studies in neuro science leading to the conclusion that the brain uses similar features in the visual cortex:

Experiments were made with micro electrodes placed in brains of cats. John and Palmer [JP87] (besides others) measured the frequency of the response to a dot-like stimulus on a homogeneous screen in a small area, the so called 'simple cells' situated in the primary visual cortex (V1). The left side of figure 3-1 shows their measurements in the space domain. Fitted with the Gabor filter in the middle, the correspondence, and as result the plausibility of the Gabor filter model is meant to be proved. In this model, visual information in space (the receptor field at the retina) as well as over time (optical flow) is represented and processes in the brain in form of Gabor filters [Wür95]. Nevertheless, there are still debates on the validity of this model.

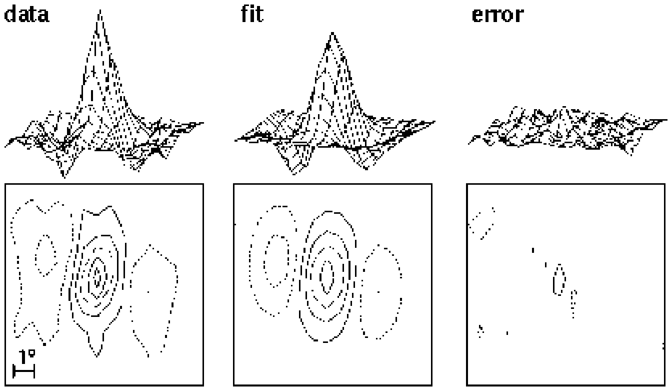

Figure 3-1: Gabor filter fitted on simple cell response (Taken from [JP87])

A Gabor kernel ψ_j is a plane sinusoids restricted by a Gaussian envelope function:

$$\psi_j(\vec{x}) = \frac{\left\|\vec{k_j}\right\|^2}{\sigma^2}\, e^{-\frac{\left\|\vec{k_j}\right\|^2 \|\vec{x}\|^2}{2\sigma^2}} \left[e^{i\langle\vec{k_j},\vec{x}\rangle} - e^{-\frac{\sigma^2}{2}} \right] \tag{3.1}$$

where $\vec{k_j}$ is the wave vector

$$\vec{k_j} = \begin{pmatrix} k_{jx} \\ k_{jy} \end{pmatrix} = \begin{pmatrix} k_\nu cos(\phi_\mu) \\ k_\nu sin(\phi_\mu) \end{pmatrix} \tag{3.2}$$

depending on the frequency ν and the orientation ϕ_μ

$$k_\nu = 2^{-\frac{\nu+2}{2}\pi} \qquad\qquad \phi_\mu = \mu\frac{\pi}{n} \tag{3.3}$$

The orientation ϕ_μ is already defined as a fraction of π so that we actually have not only one Gabor kernel but a set of n kernels with different orientations (evenly divided semicircle).

Besides the orientation there are two more parameter that determine the shape of the Gabor kernel: σ that controls the width of the Gaussian envelope and the frequency ν, in discrete images given in pixels and usually starting with the Nyquist sampling frequency. σ usually is set in a relation to the frequency.

By varying the orientation and frequency, means by varying $\vec{k_j}$ we get a family of kernels that build a so-called *jet*. Such a jet is convoluted with a small patch of gray values in an image $\mathcal{I}(\vec{x})$ around a given pixel $\vec{x} = (x, y)$

$$\mathcal{J}_j\,(\vec{x}) = \int \mathcal{I}(\vec{x}')\ \psi_j(\vec{x} - \vec{x}')\ d\vec{x}' \tag{3.4}$$

This is known as a wavelet transformation because the family of kernels is self-similar, as all kernels can be generated from one mother wavelet by dilation and rotation.

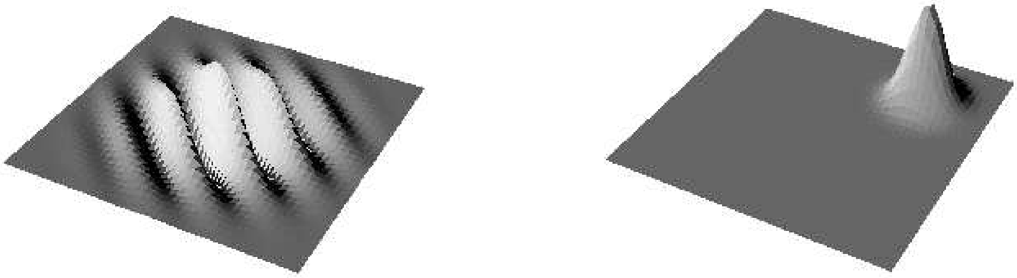

Figure 3-2: Gabor filter in space (left) and frequency (right) domain (Taken from [Pöt96])

One characteristic of wavelets is that they can be located somewhere between the space and the frequency domain. Gabor wavelet filters can be represented as Gaussian windows in the frequency domain, as can be seen in figure 3-2. The different family members or jet elements can therefore be thought of as Gaussians covering the frequency domain. This is crucial to understand their behavior as features in image processing. GW jets can be used as condense representation of a local gray value distribution (the patch), robust with respect to illumination variations, scaling, translation, and distortion. To achieve these invariances, the phase has to be removed by using the magnitude [WvMW04], since already a small displacement may lead to very different coefficients.

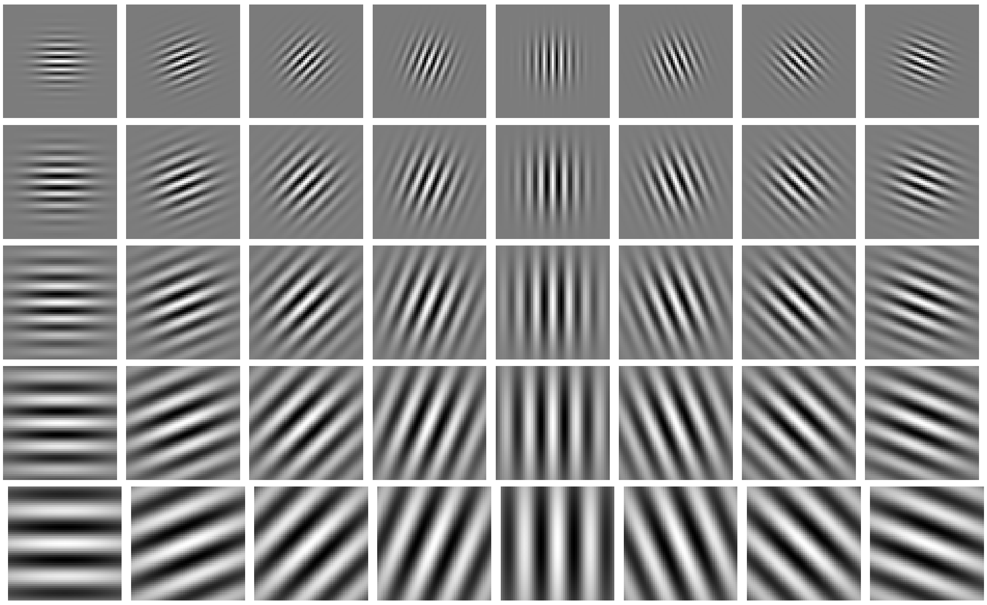

Figure 3-3: Gabor wavelet kernels of one jet in space domain (real part only)

3.2 Gradient Orientation Histogram (GOH)

Feature extraction by GOHs can not refer to a neuro-informatics background. The only relative they maybe could point to are SIFT features, used for object recognition [KS04]. The coefficients for classification in their case simply consist of a histogram over $\mathcal{G}$, the gradient orientation in a patch around a point of interest $\vec{x}$.

$$\mathcal{G}\left(\vec{x_i}\right) = \arctan\left(\frac{\frac{\partial \vec{x_i}}{\partial x}}{\frac{\partial \vec{x_i}}{\partial y}}\right) \tag{3.5}$$

where the derivative practically are done with a Sobel filters.

It was found, that a histogram with 16 bins is sufficient. It is evident that the computational costs for these features are far smaller than for applying a whole GW jet. Though, the results in our test show them performing better than GW (for details see table 5.1). This is probably due to their even higher shift insensitivity compared to GWs.

3.3 Normalization

Most existing FER systems are limited to recognizing expressions in frontal view. This limitation results from the change in appearance of the face and facial features under changing viewpoint. For instance both, GWs and GOHs depend on orientation information and are sensitive against in-plane rotations of the head. But also scaling and distortion due to out-of-plane rotations and movements towards (or away from) the camera will affect the features and decrease the classification quality.

The tracker developed in chapter 2 mainly is responsible to localize the face and to place the feature points on the same spots on the face in all images of the sequence. Furthermore, it supplies the 3d orientation of the head pose that can be used to compensate for perspectivel deformation. Applying the algorithm introduced in section 2.2.3, the patches can be warped to a state "how they would look" under another viewpoint respective another pose. A perspective normalization was done by transforming all patches to the canonical frontal view.

Changes in lighting also have an influence on the features. Histogram equalization is a simple and effective method to reduce this effect, but just calculating it for the whole image would bring up another problem in exchange: the background and hair would affect the result of the operation. Here again, information provided by the head tracker was used. Points in the 3d model were declared that surrounded the part of the face the features are placed in. Projecting theses according to the pose found by tracking they build a polygon defining the ROI for the histogram equalization. However, since special attention was given to homogeneous lighting during the recording of the test data, no improvement could be observed using this function.

3.4 Classification

Over the years many kind of classifier / feature combinations were tested in facial expression recognition. Since features and classifier build a team, they can not be compared independently. For instance, Hidden Markov Models (HMM) have advantages in classifying signals over time but require much more training data than e.g. Support Vector Machines (SVM).

As brought out earlier, in comparative studies, GW jets performed best in combination with SVMs. Based on theses results, we decided to use a SVM classifier. For this thesis, the implementation of Joachims [Joa99] was integrated in the system to perform an online classification of every frame.

Chapter 4

Facial Expression Databases

Research on facial expression recognition as on any other recognition task requires test and trainings data. This data needs to be as close to the aimed application context and situation as possible. Movellan et al. [MB] even lists comprehensive and labeled training data as the first challenge that has to be met for fully automatic facial expression measurement. This research, too, had to suffer from the very late availability of appropriate data.

Facing the same problem, different research teams answered this need with their proprietary facial databases, most of them optimized for person identification. Only some of them also contain facial expressions and just a few were collected with the intension to serve as facial expression recognition databases. In this chapter, a short survey is given about these public available databases followed by a description of a new database, compiled as part of this thesis.

4.1 Existing Facial Expression Databases

Databases used for facial expression analysis can be divided into two groups: One that is labeled according to the FACSystem (see appendix B) and one for which the main focus lies on person identification but also contains samples with different expressions. The comparison in table 4.1 gives a survey but does not claim completeness; e.g. commercial databases as the ones of Paul Ekman where ignored. More detailed information can be given for the Human identification on Distance (HID) database and the Cohn-Kanade database, as they have been evaluated for our research purposes.

Year	Name	Images	Sequences	Subjects	Comment	Ref.
2004	CBCL-MIT (in progress)	no	about 1200, RGB	12, men and women, mult. ethnics	acted and natural expressions	NA
2004	HID (in progress)	for all subj. 9 mug shots	many, RGB	284, men and women, mult. ethnics	natural expressions	NA
2000	Cohn-Kanade (DFAT-504)	no played	329, Gray, RGB	100, students mult. ethnics	FACS coded	[TKT00]
2000	PIE	more than 40,000	yes, but for talking only	68	different pose, illum. and expr.	[SBB03]
1997	FERET	Gray, RGB(new)	No	1009	designed for identif.	[PRD96]
1998	JAFFE	213 B/W B/W	no	10 Japanese woman	basic emotions played	NA

Table 4.1: Survey of Facial Espression Databases (*in progress)

4.1.1 Cohn-Kanade database

Cohn, Kanade and Tian published 2000 a database that can be considered as todays de-facto standard for comparative studies on facial expression analysis. It is totally FACS orientated since not only the labeling is done according to the FACSystem but also the subjects were instructed by the experimenter to perform specific single AUs or combinations of these. All desired displays of AUs were described and modeled prior to recording by the research scientist.

The database consists of sequences of 9 to 60 frames, where each frame is stored as single image in PNG format. Sequences in the available version start at neutral and end at the maximum intensity of the performed AU, though they were recorded originally until the neutral state was reached again. 100 students between 18 and 30 years old volunteered. 65% were female, 15% were African-American and 3% were Asian or Latino. Two cameras were used, one in frontal view and one under 30°, but currently only the data in frontal view are published.

Heads in the available portion of the database are all shown in frontal view with almost no motion. This is an advantage for most purposes, but unfortunately it is useless to evaluate the head tracker developed in this thesis. Another disadvantage of this database is the image quality. Some sessions were over exposed or affected by interlace effects.

The system that was developed in this theses was meant to classify facial expressions into the six basic emotions. Since all labeling was done in AUs, the database is not suitable for this task. The authors provided a 'set of translation rules' for the AU codes to emotion labels but advices at the same time: "There are many reasons for being cautious in the use of this table". Anyway, "dozen of minor variants for each of the emotion" make it impossible to automate the training and testing. However, to compare the GOH features as introduced in this thesis to the common GW approach, we trained and tested it with this database on AU recognition instead. The results can be found in section 5.1.

4.1.2 HID database by o'Toole

Very recently, Alice o'Toole collected a database of static images and video clips for the Human Identity DARPA Project. This database was designed to test algorithms for face and person recognition and for tracking and model human motions. It includes 9 static facial mug shots per subject and video sequences showing the subject talking, walking, in a conversation and another, this time moving mug-shot. Of more interest for this research were the dynamic facial expression clips that show the natural expressions of the subject while watching a 10 minute video. This video contained scenes from various movies and television programs intended to elicit different emotions. Based on the judgment of the experimenter, facial expressions where cut into 5 second clips and labeled into happiness, sadness, fear, disgust, anger, puzzlement, laughter, surprise, boredom, disbelief or blank stare. This way, the emotions are captured under very natural conditions, including head and eye movements.

Along with these natural conditions come the drawbacks for a recognition system:

- most expressions are very subtle.
- some clips contain more than one expression (e.g. puzzled expression, which turns to surprise or disbelief and ultimately to laughter).
- the six basic emotions defined by Ekman [EF78] are not appropriate for classification, resulting in 12 classes.
- per subject and class are rarely more than two samples were recorded.

Especially the latter caused us not to use this database for our research. Despite the fact that SVM classifier need little training data, there is not enough to represent the in-class diversity due to the different subjects, head movements and blended expressions.

Besides this, the database was shipped on a macintosh formatted hard disk (with an overall amount of more than 160 GB) in an for Windows operating systems unusual DV-codec. The labeling is done in a filename and folder system that makes automated test series very complicate.

4.2 Yet Another Database: The CBCL-MIT database

Since the available databases did not match the demands of our experiments, it was necessary to build a new database (CBCL-MIT database), with a sufficient number of sample sequences of the six basic emotions for both training and testing. The aimed context was that of a human computer interaction with frontal view but natural head motions. The subjects therefore were asked to look directly into the camera placed in front of them. To obtain the desired number of sequences, each expression was repeated several times. After this part for acted expressions, the subjects watched a 20 minute movie that consisted of short video clips. These clips covered a large variety of topics from funny situations over surgery scenes to recent war documentation. Similar to the HID database the labeling of the recorded natural expressions was done by the judgment of the experimenter.

4.2.1 Experimental Setup

Like for the Cohn-Kanade database two DV cameras where used, one in frontal view and one under an angle of 30°. The camera in frontal view was a Sony Digital Handycam DCR VX2000 NTSC and the one from the side the slightly older model DCR VX1000. Besides the homogeneous ceiling light a diffused reflected (by a reflection umbrella) floodlight was placed behind the camera in frontal view at a hight of approximately 2 meters.

The video sequences were captured via the IEEE 1394 bus and cut with Adobe Premiere version 6.5. Table 4.2 lists the technical data of the resulting video clips.

Video standard	NTSC
Codec	MainConcept DV Codec 2.4.4
Audio	no audio was captured
Frame rate	29.97 fps, non drop-frame
Frame size	720 × 480
Pixel aspect ratio	D1/DV NTSC (0.9)

Table 4.2: technical data of captured video clips

Since the cameras work in interlaced mode, fast movements produce a horizontal stripe pattern where the head position is slightly shifted between the even and odd lines. This significant reduces the image quality, just like in the Cohn-Kanade database, but can be removed by common deinterlace techniques.

Another effect comes from the pixel aspect ration of 0.9 due to the NTSC standard. You will find the clips displayed automatically distortion-free by almost all video renderer with a width of 648 pixel. However, the original image data has a width of 720 and the image is horizontal stretched. This has to be considered when processing the data, since the images will keep this distortion until the final rendering step.

4.2.2 Statistical Overview

Twelve persons volunteered to participate. They ranged in age from 18 to 30 years. Eight subjects were female, three were African-American, one Asian and one Indian.

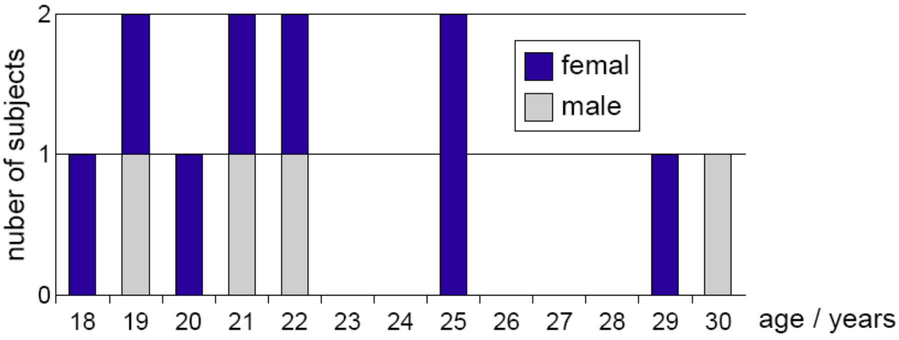

Figure 4-1: Age of subjects

Up to now, the data of 6 persons are cut and digitized. The remaining data of 6 subjects probably will be processed by the end of October 2004. For each basic emotion, an average of 11.5 sequences per subject were collected. The average number of sequences with spontaneous expressions per person is 27.8.

Chapter 5

Experimental Results

5.1 Recognizing Facial Action Units on the Cohn Kanade Database

As mentioned in section 4.1.1, all samples in the Cohn-Kanade database are labeled according to the Facial Action Coding System (FACS). Nevertheless, features that are designed for emotional classification of facial expressions also should be able to recognize Action Units (AU).

The labeling consists of a file which lists all AUs that where found by the human specialists in an image sequence of one facial expression. These sequences usually go from the neutral to the maximum state, so that the later images show the AU to detect in an increasing intensity. Therefore, from sequences with more than 12 images, the last 4 where taken into account, resp. the last 2 of sequences with 9 to 12 images and only 1 when the sequence consists of at least 6, but less than 9 images.

Most AU classifications are subdivided, some in up to 5 classes. To simplify and automate the test, a software tool was written that only considered the 'pure' Labels (like e.g. '17', but not '62d' or 'L12b') to generate lists of training and test data. By random, 50% of the sample images were choosen as training and 50% as test data. The 5 AUs that have been selected for this test qualified by a simple fact: They are the only AUs labeled on more than 100 sequences each, allowing for a sufficiently large number of training and test samples. In Detail, this were

- AU 1: Inner Brow Raiser

- AU 4: Brow Lowerer
- AU 6: Cheek Raiser
- AU 17: Chin Raiser
- AU 25: Lips part

Unlike the features that can be expected to handle this slightly different task, the face tracker as presented in chapter 3 is not self-evident a appropriate mean for face detection. Equipped only with the feature templates of one sample image, means one person, it still performed very well considering the variety due to gender, make-up, skin colors and facial hair. Sample images that were not detected with the necessary accuracy where deleted from the training and test data list by hand. The detection rate across people ranged from 70.8% to 93.3%, the average was 83.7% subjectively correct found head poses.

Since several AUs go along to express one emotion, their appearance is highly correlated. Though one feature set covering the whole face therefore would bring up better recognition results, to evaluate the detection of each AU separately, the features had to be set around the location on the face, where the AU occurs. This results in different feature sets for every AU. The alignment was fist optimized for the GOH-Features and then again for the GW-Features. Figure 5-1 shows the feature placement for all AUs. Besides this, it gives a good survey of the test, since every row consists of (from left to right) one image with correct classified AU (by GOH) followed by two samples where the AU was not detected, although FACS coder labeled it on this image. The image in the middle is misclassified using GOH features and the right one using GWs.

Unfortunately, only 11 of the total 138 recorded persons allowed to use their images for publication. Amazingly, the image quality of these sessions are all above the average of this database. Therefore is not possible to present samples whiche were affected by overexposure or the interlaced mode of the camera. That is also the reason why there is an empty space in middle of figure 5-1: For none of the misclassified images in this section was a permission given for publication.

For the GOH features, a histogram for each feature point consists of 16 directions. The Gabor Wavelet Jet consists of 5 orientations and 8 frequencies (following the example of [YTC02, DBH$^+$99, DGA00]), this are 40 coefficients per feature vector.

	AU 1	**AU 4**	**AU 6**	**AU 17**	**AU 25**
Correct detected					
Gabor Wavelets	89.10%	85.85%	91.38%	89.80%	92.96%
Gradient Orientation Hist.	93.58%	91.18%	93.22%	94.90%	93.09%
Number of Features					
Gabor Wavelets	12	11	6	11	7
Gradient Orientation Hist.	10	9	6	9	7
Number of Feature coefficients					
Gabor Wavelets 5 frequencies and 8 directions	12*5*8 = 480	11*5*8 = 440	6*5*8 = 240	11*5*8 = 440	7*5*8 = 280
Gradient Orientation Hist. with 16 bins	10*16 = 160	9*16 = 144	6*16 = 96	9*16 = 144	7*16 = 112
Feature Size in Pixel					
Gabor Wavelets	20x20	20x20	20x20	20x20	25x25
Gradient Orientation Hist.	18x18	18x18	25x25	25x25	25x25

Table 5.1: Comparison Gabor-Wavelets vs Gradient-Orientation-Histogram

	AU 1	**AU 4**	**AU 6**	**AU 17**	**AU 25**
Test images positive	190	202	148	193	447
Test images negative	480	512	560	493	306
Training images positive	182	204	155	196	419
Training images negative	468	471	526	506	299

Table 5.2: Test and training data

All patches were warped to a normal view by default as explained in section 2.2.3, though head rotations are minimal in this database. Since the distance od the faces from the camera ranged from about 1.0 to 1.6 meters, a standard distance of 1.8 meter was chosen for the tracker:

$$p_{Norm} = \{\ x = 0\ mm,\ y = 0\ mm,\ z = 1800\ mm,\ \alpha = 0.0^\circ,\ \beta = 0.0^\circ,\ \phi = 0.0^\circ\} \tag{5.1}$$

Besides this, no special preprocessing was used for the GOH, expect transforming the image to gray scale in case of color images.

5.1.1 Comparison to Results Published by Tian, Kanade and Cohn

Tian, Kanade and Cohn [YTC02] evaluated Gabor Wavelets for Facial Action Unit recognition on their database. They placed 20 feature points in the upper face part and calculated the coefficients for 5 frequencies and 8 direction. The resulting coefficients were used to support their already existing system based on geometric features.

	Gabor Wavelets		**Gradient Orientation Hist.**		**GW & geom. f. by Tian et al.**	
	correct positive classified	false positive classified				
	false negative classified	correct negative classified				
AU 1	122	5	150	3	101	4
	68	475	40	477	3	91
AU 4	101	0	152	13	75	11
	101	512	50	499	9	104
AU 6	97	10	116	16	45	7
	51	550	32	544	7	140
AU 17	132	9	166	8	NA	
	61	484	27	485		
AU 25	426	32	431	35	NA	
	21	274	16	271		

Table 5.3: Comparison table confusion matrices

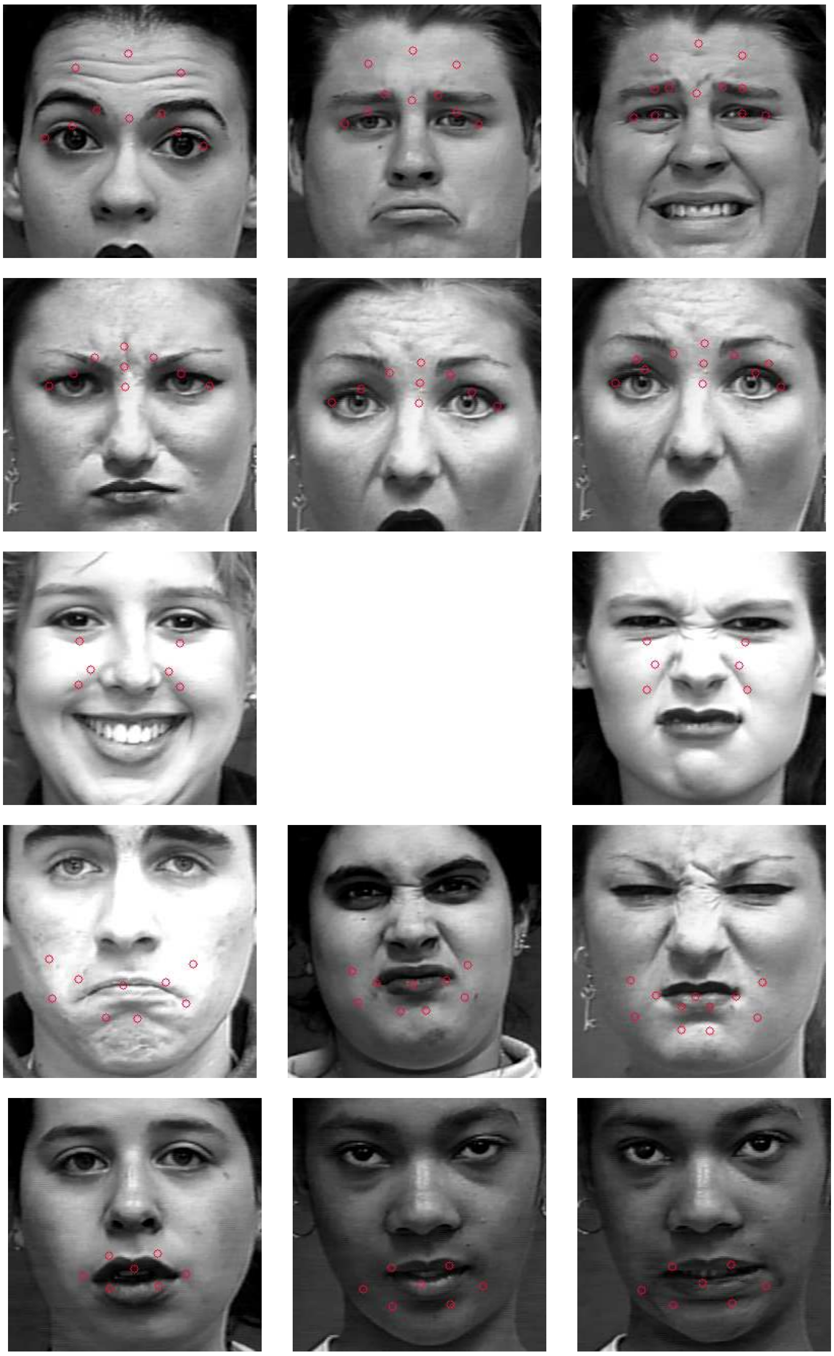

Figure 5-1: Sample images from the test set. From left to right: GOH - correct positive, GOH - false negative (AU missed), GW - false negative (AU missed). From up to down: AU 1 (Inner Brow Raiser), AU 4 (Brow Lowerer), AU 6 (Cheek Raiser), AU 17 (Chin Raiser), AU 25 (Lips part)

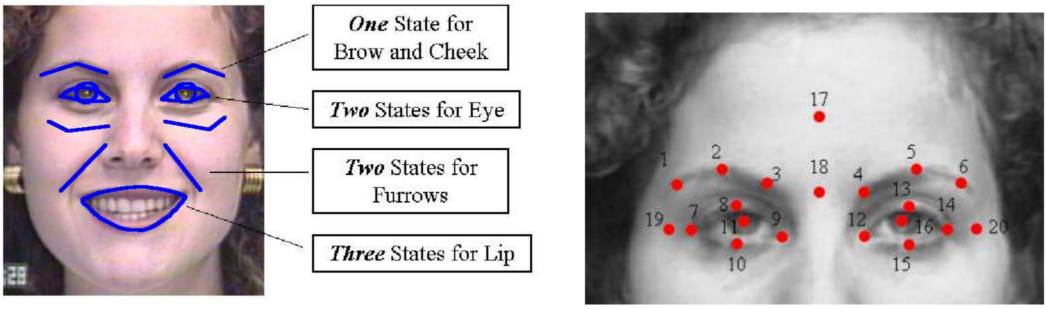

Figure 5-2: The placement of geometic and GW features used by Tian et al.[YTC02].

For a fair comparison, it has to be considered that their system differs from the one used in this thesis in various ways (see table 5.5). For instance, instead of SVM a three layer neural network served as classifier with both input from Gabor Wavelets and geometric features. While in this thesis single images where analyzed, Tian et al. computed one result over a whole sequence of images. They worked with the same database, yet not all of this database is published. Therefore the test and training data might differ in parts. Hence they included AU 1, 4 and 6 in their test, it is possible to compare the final recognition rates. Considering the effort and computational costs, the combination of GOH with SVM can be said to outperform the combination GW with SVM as well as GW and geometrical features in conjunction with a neural network (see table 5.5).

	Tian, Kanade and Cohn	**this thesis**
Distribution of the data pool	2/3 training and 1/3 test data	1/2 training and 1/2 test data
Amount of samples used for test per AU	all the same 199 sequences	between 670 and 708 images
Feature sets	one for all	each AU has it's own optimized set
Classifier	Neural Network	SVM
Classification based on	GW and geometric features whole sequences	either GW or GOH single images

Table 5.4: Comparison between the system published by Tian et al [YTC02] and the one used in this thesis

	AU 1	**AU 4**	**AU 6**
Gabor Wavelets	89.10%	85.85%	91.38%
Gradient Orientation Hist.	93.58%	**91.18%**	**93.22%**
GW & geom. feat. (Tian et al.)	**96.48%**	89.95%	92.96%

Table 5.5: Comparison of recognition rates (for details see table 5.3)

5.2 Performance

The objective of this work was not a real-time processing. However, one advantage of the system presented in this thesis is its relatively small computational cost.

The performance was measured on a short sequence, including normal head movement. The sequence was played forwards and reverse for several times, so that each frame was processed 500 times. Only the program segments for pose tracking and feature extraction were taken in account, all other modules, esp. drawing functions, were deactivated. The sequence was completely loaded to RAM and no image data was displayed. Templates for tracking were updated for every frame, although in normal operation a minimum pose difference limit drastically reduces the number of calls for this function. Ten patches for feature extraction are warped and successive GWs and GOHs applied on them. Tests were made on a Laptop equipped with a 1.06 GHz Intel Pentium III and 512 MB RAM.

With 80,6% of the computation time, the normalized cross correlation (NCC) for feature tracking is the most expensive function. Downsampling and histogram equalization are next on the list with 4.5% and 4.1%, respectively. Feature extraction with GW takes 3.1% compared to 1.5% for the GOH. Warping for both tracking and feature extraction only takes 2.1% of the overall computation time. The portion of the 3d model pose parameter estimation are less than 0.1%.

Table 5.6 and 5.7 list some absolute values. All measurements are list in milliseconds.

	downsampling	**NCC for 1. step**	**NCC for 2. step**
time / ms	5.299	1.862	56,308

Table 5.6: Computation time for tracking

To evaluate these results, they have to be compared to the speed of spontaneous facial expressions. A system that is meant for natural interaction with humans would be expected not to miss even short expressions. In section 1.1 a statement was cited, claiming facial expressions "lasting [...] rarely [...] less than 250 ms" [FL03]. With this as reference, the

	patch size / pixel			
	15×15	20×20	25×25	30×30
warping	0.915	1.494	2.236	3.152
GW	1.281	2.059	3.014	4.087
GOH	0.608	1.156	1.882	2.790

Table 5.7: Computation time for feature extraction in milliseconds

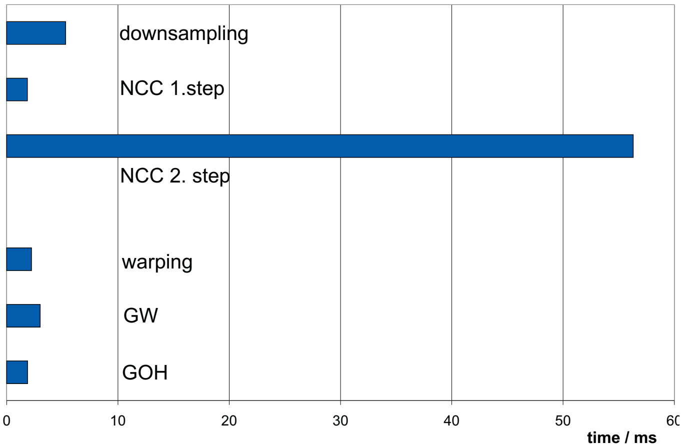

Figure 5-3: Comparison of computation time (top: tracker, bottom: feat. extr. on 10 20×20 patches)

results presented here can be considered to be good enough regarding speed, although the accuracy esp. of the head tracker is yet not sufficient for a real world application. However, these data clearly show that a 3d model-based viewpoint normalization can be done in real-time.

Unfortunately, we did not have the appropriate test data, to measure the angle up to which the system is able to recognize facial expression with a satisfyingly accuracy.

Chapter 6

Conclusion

This research addressed the problem of automatic facial expression recognition. A computer vision system was developed that automatically tracks the 3d head pose and warps small patches of the face to a normalized canonical frontal view. On these patches either Gabor wavelet (GW) jets or gradient orientation histograms (GOH) were computed to extract information about the displayed expression. A Support Vector Machine then was used to classify the expression.

Test and Training data was obtained by recording a database of played and natural expressions from the same subjects. For each of the basic emotions there are about 11.5 sequences in average, further about 27.8 sequences of spontaneous expressions, building a sufficient base for training most types of classifiers. This database already is used to explore other approaches for facial expression recognition.

In tests, the pose tracker did not fully meet expectations but performed good enough for the purpose it was designed for. Further, by comparing our system with published results of another research group, GOH outperformed commonly used GW jets.

6.1 Suggestion for Future Work

When analysing facial expressions, humans always consider context information (if available): the situation, knowledge about the observed person, speech, voice, hand and body gestures. In a similar way, an automatic system would need to obtain and combine information from different cues. For a reliable emotion interpretation in human computer interaction, facial expression recognition therefore can only be considered one part or module of a holistic approach.

But also from the face itself, more information could be extracted. Different research groups explore the dynamics of expressions. Eye movements and head pose are important means to communicate emotional states like boredom or interest and to emphasize expressions. Including these in the analysis, probably a wider range of expressions can be read from the face and distinguished.

Recording the facial expression database, we found that the 6 basic emotions as introduced in the 70ies are not suitable to represent spontaneous expressions observed on our subjects. Especially concerning human computer interaction, they will not build a adequate base for classification. We see a need for research to define classes of emotional states which depend on the situation. For instance for user interface devices, puzzlement and impatience might be more relevant emotional categories than fear.

Appendix A

Details of the Implementation

The automatic facial expression recognition system presented in this thesis was fully implemented in C++ using Microsoft VisualC++ 6.0. The video processing is based on Microsoft DirectX SDK 9.0b (with a project wizard of Yunqiang Chen) and the interface uses MFC. Of the OpenCV library only drawing functions and once a matrix inversion remained in the project, after a closer look discovered some memory leaks in other functions used before. The support vector machine used for classification is the SVMlight implemented by Thorsten Joachims [Joa99]. All other implementation was done by the author.

The system is divided into 3 main parts or layers: On the top the interface, in the middle the DirectShow video streaming and at the ground the EmAn.lib containing the data processing. This clear separation has proofed to keep systems more flexible to adapt to other interfaces (e.g. used for the tests with the Cohn Kanade database, see A.4) or even different operating system environments (Linux does not yet support DirectX).

Figure A-1 shows this structure in more detail. The blue arrows represent the way the video data takes through the application. DirectShow provides the source data, whether reading from a file or a live source like a camera. It then decompresses and transforms it if necessary until it matches the input format expected by the EmAnF.ax filter. When passing this filter, each frame is given to the processing unit in EmAn.lib. Afterwards it is rendered by DirectShow in the application window. Red arrows show the command and information paths. The application builds and controls the DirectShow filtergraph, a COM-object wrapping all filters in the streaming chain, and through the EmAnF.ax filter the processing itself. The processing unit in return gives information like the pose

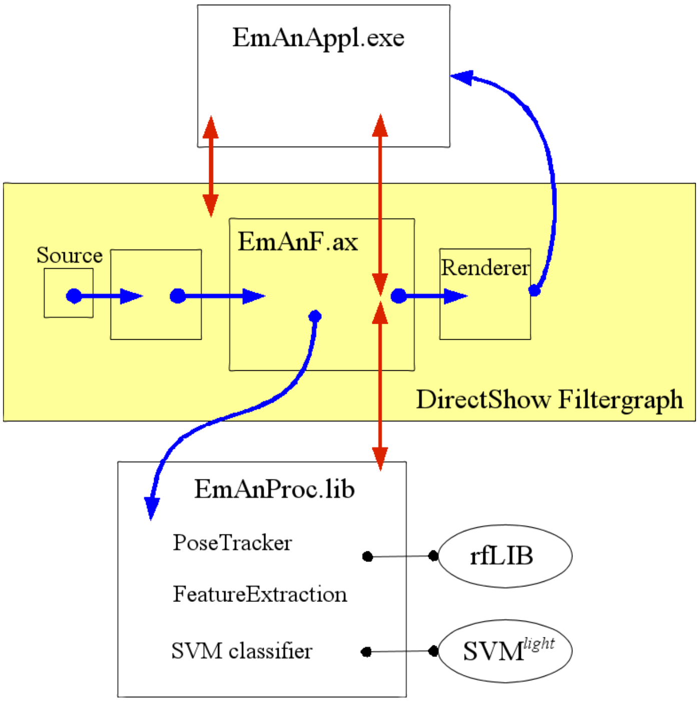

Figure A-1: System survey

parameters or the processing duration to the application via the filter.

A.1 EmAn.lib: The image processing unit

Figure A-2 provides a reduced UML diagram of the processing unit. The *EmAnProc* class hosts mainly three objects (orange marked): the first one for pose tracking (*PoseTracker*), the second one for Feature extraction (*GWreader*) and the third is a wrapper class for the SVMlight functions. Beside these it provides some utilities like the histogram equalization (see section 3.3), saving single frames as image file or performance measurement and is responsible for the communication with the Application (via *EmAnDataPipe*).

PoseTracker and *GWreader* own feature objects (blue marked), the first one specialized for tracking and the later one for extracting the data used for the classification. Since these have a lot in common, they are both inherited from the *Feature* class, where e.g. the warping is done (see section 2.2.3). On the other hand, e.g. the NCC is only used by features to track and therefore implemented in *TrackFeature*. Lowes algorithm for parameter estimation (see section 2.2.1) can be found in the *JacPoseFinder* module. *GaborJet2d* holds the jet consisting of complex GWs as an array of floating images and performs the calculations of the coefficients (see equation 3.4).

The *Pose* and *FeatPoint* modules are smaller data structures that hold an important part of the preprocessing. Many operators are overloaded for them to simplify their handling. The task for *ClfIn* is to collect, manage and log the results of the *PoseTracker* and *GWreader* and to provide them to the classifier.

The *SVMclassifier* manages the models, prepares the data for classification and logs the results. Some small changes had to be made on the original source code of the SVM itself to fit the C-routines in a C++ environment, esp. regarding memory and error handling.

The structure *EmAnDataPipe* is used to pack the information that are send through the filter to the application.

A.2 EmAnF.ax: DirectShow for video streaming

DirectShow as a part of DirectX is the standard mean on windows operating systems to process and display video streams. Due to its architecture, it simplifies the task of reading different video formats or addressing a live video source. Since it is part of the operating system and very machine-intimate, its display functions take little of the computer resources. On the other hand, it designed rather for multimedia entertainment applications than for scientific image processing.

To ensure the full control over the processing, it was necessary to write a new DirectShow filter module derived from the *CTransformFilter* base class (provided by the DirectX SDK). This object in Windows COM+ design can be directly integrated in the filter chain for video streaming. All frames are first given to it and have to path the function `Transform(IMediaSample *pMediaSample)` before being rendered on the display. In this key function, all image processing is linked in. Depending on the filter state, set by the application, the tracking and classification can be switched on and off or the modules reseted, the current frame can be saved as Bitmap, the selection for a tracking feature be set or all processing be deactivated.

The filter communicates with the application by windows messages and its interface *IEmAnF*, another COM+ object owned by the application.

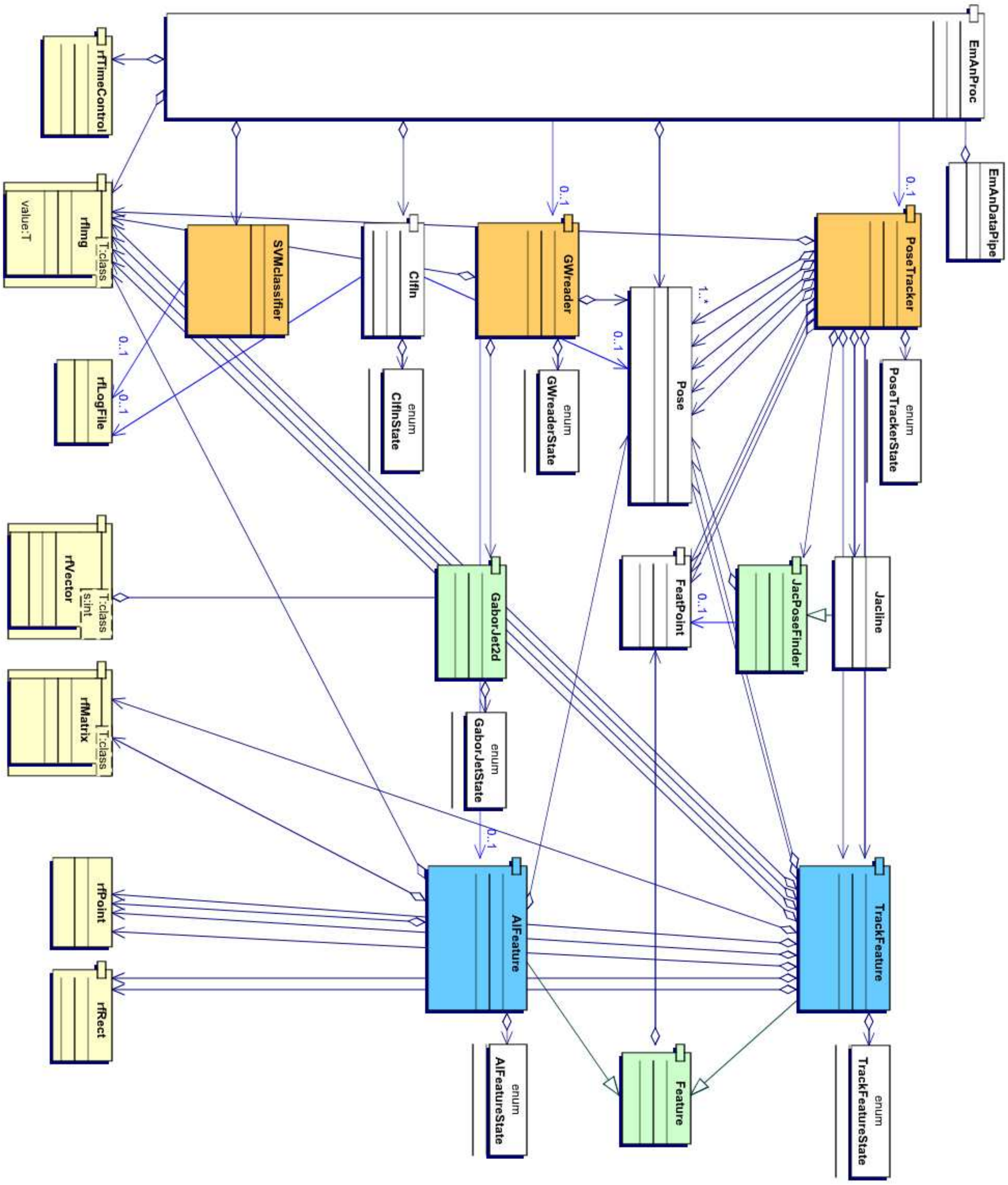

Figure A-2: EmAnProc UML survey

A.3 EmAn.exe: Interface

The graphical interface is meant to simplify the initial feature selection and to check the processing results. It also provides a toolbar for convenient labeling of test and training data. Besides this, this unit holds and controls all parts that are used for the video streaming.

A.4 CoKa.exe: EmAn test program for the Cohn-Kanade database

This module was written to perform the test on the Cohn-Kanade database (see section 5.1). It is a simple console application that primarily generated and sorted the file lists of the face images and split them randomly into test and trainings data. The face detection and pose finding followed by a manual verification was done one time for the database. With these informations the different test programs were run on the whole database. Each test program running in the background took about 20 minutes on a 1Ghz Laptop.

Appendix B

Facial Action Coding System

Paul Ekman and W.V. Friesen [EF78] developed in the 1970s[1] the Facial Action Coding System (FACS) based on how the contraction of each facial muscle (singly and in combination with other muscles) changes the appearance of the face. The system was designed to identify the specific changes that occurred with muscular contractions and to describe them a way that best differentiate one from another. Their goal was to create a reliable means for skilled human scorers in human behavior psychology to determine the category or categories in which to fit each facial display.

Using the anatomic model of Hjortsjö, Ekman and Friesen declared 44 so called Action Units (AU) to code a facial appearance. Though 30 of them are anatomically related to a specific set of facial muscles, AUs can not be directly translated to muscles or their contraction. Some of them are performed by more than one muscle and again other AUs represent relatively independent actions of different parts of the same muscle. For the remaining 14, like e.g. 'Bite lip' (32), the anatomic basis is even unspecific. Rather they can be seen as the elements, in which a behavioral psychologists decomposes an observed expression. The result of encoding a facial expression in the FACS consist of a list of AUs that produced it.

The crucial point about the FACS is its clear devision between a description of really observed appearance of the face and it psychologically interpretation. Other concepts such as FACSAID (Facial Action Coding System Affect Interpretation Dictionary) are applied following the 'encoding' to links facial expressions with their psychological interpretations.

[1] In 2002, Ekman published a new, revised version of the FACS.

FACS coding is currently performed by trained experts who make perceptual judgments of video sequences, often frame by frame. It requires approximately 100 hours to train a person to make these judgments reliable and pass a standardized test for reliability. It then typically takes over two hours to code comprehensively one minute of video [BML$^+$].

Automated systems would not only tremendously reduce this effort but also could reveal characteristics of dynamics in different AUs. Groups like the one of Jeff Cohn and Takeo Kanade, Marian S. Bartlett, Maja Pantic and others therefore focus their research on techniques for FER that perform this coding automatically for behavioral sciences.

Bibliography

[BBLM01] B. Braathen, M. Bartlett, G. Littlewort, and J. Movellan. First steps towards automatic recognition of spontaneous facial action units. In *Proc. of the 2001 workshop on Percetive user interfaces (Poster Session)*, 2001.

[Bel96] Sir C. Bell. *Essays on the Anatomy of Expression in Painting.* Longman, Reese, Hurst & Orme, London, third edition, 1896. first edition published 1806.

[BKP03] I. Buciu, C. Kotropoulos, and I. Pitas. Ica and gabor representation for facial expression recognition. In *Proc. of Int. Conf. on Image Processing*, volume 2, pages 855–858, 2003.

[BLB$^+$03] M. Bartlett, G. Littlewort, B. Braathen, T. Sejnowski, and J. Movellan. *A prototype for automatic recognition of spontaneous facial actions.* 2003.

[BLFM03] M. Bartlett, G. Littlewort, I. Fasel, and J. Movellan. Real time face detection and expression recognition: Development and application to human-computer interaction. In *CVPR Workshop on Computer Vision and Pattern Recognition for Human-Computer Interaction*, 2003.

[BML$^+$] M. Bartlett, J. Movellan, G. Littlewort, B. Braathen, M. Frank, and T. Sejnowski. *Towards Automatic Recognition of Spontaneous Facial Actions.*

[Dar72] C. Darwin. *The Expression of the Emotions in Man and Animal.* J. Murray, London, 1872.

[DBH$^+$99] G. Donato, M.S. Barlett, J. Hager, P. Ekman, and T. Sejnowski. Classifying facial actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):974–989, 1999.

[DGA00] N. Dailey, G.Cottrell, and R. Adolphs. A six-unit network is all you need to discover happiness. In *Proc. of the 22 Annual C. of the Cognitive Science Society*, 2000.

[EF71] P. Ekman and W. Friesen. Constants across cultures in the face and emotion. *Journal of Personality and Social Psychology*, 17(2):124–129, 1971.

[EF78] P. Ekman and W. Friesen. *The Facial Action Coding System.* Consulting Psychologists Press Inc., Palo Alto, Calif., 1978.

[EG98] P. Eisert and B. Girod. Analysing facial expressions for virtual conferencing. *IEEE Computer Graphics and Applications*, pages 70–78, September 1998.

[EP95] I. Essa and A. Pentland. Facial expression recognition using a dynamic model and motion energy. In *ICCV*, pages 360–367, 1995.

[FL03] B. Fasel and J. Luettin. Automatic facial expression analysis: A survey. *Pattern Recognition*, 36(1):259–275, 2003.

[FT03] D. Fidaleo and M. Trivedi. Manifold analysis of facial gestures for face recognition. In *Proc. of the 2003 ACM SIGMM workshop on Biometrics methods and appl.*, pages 65–69, 2003.

[HTF01] T. Hastie, R. Tibshirani, and J. Friedmann. *The Element of Statistical Learning.* Springer, 2001.

[Joa99] T. Joachims. *Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning.* MIT-Press, 1999.

[JP87] J. Jones and L. Palmer. An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. *J. Neurophysiol.*, 58(6):1233–1258, 1987.

[KB03] D. Kim and Z. Bien. Fuzzy neural networks (fnn)-based approach for personalized facial expression recognition with novel feature selection model. In *Proc. of IEEE Int. Conf. on Fuzzy Systems*, pages 908–913, 2003.

[KQP03] A. Kapoor, Y. Qi, and R. Picard. Fully automatic upper facial action recognition. Technical Report 571, MIT Media Lab, Affective Computing Technical Report, 2003.

[KS04] Y. Ke and R. Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. *Computer Vision and Pattern Recognition*, 2004.

[LAKG98] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba. Coding facial expressions with gabor wavelets. In *Proc. third IEEE Int. C. on Automatic Face and Gesture Recognition*, pages 200–205, 1998.

[Lev44] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quart. Appl. Math.*, 2:164–168, 1944.

[LHO00] G. Loy, E.J. Holden, and R. Owens. A 3d head tracker for an automatic lipreading system. In *Proc. of Australian Conference on Robotics and Automation (ACRA2000)*, 2000.

[Low91] D.G. Lowe. Fitting parameterized 3-d models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441–450, May 1991.

[LTC97] A. Lanitis, C. Taylor, and T. Cootes. Automatic interpretation and coding of face images using flexible models. In *IEEE Transaction on Pattern Analysis and Machine Intelligence*, pages 743–746, 1997.

[LVB$^+$] M. Lades, J. Vorbruggen, J. Buhmann, J. Lange, C. v.d. Malsburg, and R. Würtz.

[Mar63] D.W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal Soc. Indust.Applied Math*, 11(2):431–441, 1963.

[MB] J. Movellan and M. Bartlett. *The Next Generation of Automatic Facial Expression Measurement.*

[OO98] T. Otsuka and J. Ohya. Recognizing abruptly changing facial expressions from time-sequential face images. In *Proc. of Computer Vision and Pattern Recognition Conf. (CVPR98)*, 1998.

[OPB97] N. Oliver, A. Pentland, and F. Berard. Lafter: Lips and face real time tracker with facial expression recognition. In *Proc. of Computer Vision and Pattern Recognition Conf. (CVPR97)*, 1997.

[PA96] P. Penev and J. Atick. Local feature analysis: a general statistical theory for object representation. *Computation in Neural Systems*, 7(3):477–500, 1996.

[Pöt96] M. Pötzsch, 1996. Internet publication of the University of Bochum.

[PR98] M. Pantic and L. Rothkrantz. Automated facial expression analysis. In *Proc. of the fourth annual conf. of the Advanced School for Computing and Imaging(ASCI'98)*, 1998.

[PR00] M. Pantic and L. Rothkranz. Automatic analysis of facial expressions: The state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1424–1445, 2000.

[PRD96] P. Phillips, P. Rauss, and S. Z. Der. Feret (face recognition technology) recognition algorithm development and test results. *Army Research Lab technical report*, (995), 1996.

[SBB03] T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination, and expression database. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12):1615–1618, 2003.

[SGG02] C. Tomasi S.B. Goturk, J.-Y. Bouguet and B. Girod. Model-based face tracking for view-independent facial expression recognition. In *Proc. of the Fifth IEEE Int. C. on Automatic Face and Gesture Recognition (FGR'02)*, 2002.

[SP92] A. Samal and P.Iyengar. Automatic recognition and analysis of human faces and facial expressions: A survey. *Pattern Recognition*, 25(1):65–77, 1992.

[TKT00] J. Cohn T. Kanade and Y. Tian. Comprehensive database for facial expression analysis. In *The 4th IEEE Int. C. on Automatic Face and Gesture Recognition (FG'00)*, 2000.

[WFKvdM99] L. Wiskott, J.-M. Fellous, N. Krüger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. In L. C. Jain, U. Halici, I. Hayashi, and S. B. Lee, editors, *Intelligent Biometric Techniques in Fingerprint and Face Recognition*, chapter 11, pages 355–396. CRC Press, 1999.

[Wür95] R. Würtz. *Multilayer Dynamic Link Networks for Establishing Image Point Correspondences and Visual Object Recognition.* Verlag Harri Deutsch, Thun, Frankfurt am Main, 1995.

[WvMW04] I. Wundrich, C. v.d. Malsburg, and R. Würtz. Image representation by complex cell responses. *Neural Computation*, 16(12), 2004. In press.

[YTC02] T. Kanade Y. Tian and J. Cohn. Evaluation of gabor-wavelet-based facial action unit recognition in image sequences of increasing complexity. In *Proc. of the Fifth IEEE Int. C. on Automatic Face and Gesture Recognition (FGR'02)*, pages 229–234, 2002.